UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/087,296	03/01/2002	Anil Seth	1488.008US1	2127

21186   7590   01/03/2007
SCHWEGMAN, LUNDBERG, WOESSNER & KLUTH, P.A.
P.O. BOX 2938
MINNEAPOLIS, MN 55402

EXAMINER
ROMANO, JOHN J

ART UNIT	PAPER NUMBER
2192	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	01/03/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

PTOL-90A (Rev. 10/06)

	Application No.	Applicant(s)
Office Action Summary	10/087,296	SETH ET AL.
	Examiner	Art Unit	
	John J. Romano	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
  Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on <u>*16 October 2006*</u>.

2a)☒ This action is **FINAL**.   2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-44* is/are pending in the application.

   4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-44* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

   Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

   Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

   a)☐ All b)☐ Some * c)☐ None of:

   1.☐ Certified copies of the priority documents have been received.

   2.☐ Certified copies of the priority documents have been received in Application No. _____.

   3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

   * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
    Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1.   Applicant's amendment and response received October 16th, 2006 responding to

the June 15th, 2006, Office action provided in the rejections of claims 1-44, wherein

claims 1-99 are pending in the application and which have been fully considered by the

examiner.

Applicant arguing for the claims being patentable over *the prior art* (see pages

12-17 of the amendment and response) are not persuasive, as will be addressed under

Prior Art's Arguments – Rejections section at item 2 and the claim rejections below.

Accordingly, Applicants' arguments necessitated additional clarifications. Thus, the

rejection of the claims over prior art in the previous Office action is maintained in light of

the necessitated additional clarifications provided hereon and **THIS ACTION IS MADE**

FINAL. Applicant is reminded of the extension of time policy as set forth in 37

CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE

MONTHS from the mailing date of this action. In the event a first reply is filed within

TWO MONTHS of the mailing date of this final action and the advisory action is not

mailed until after the end of the THREE-MONTH shortened statutory period, then the

shortened statutory period will expire on the date the advisory action is mailed, and any

extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later

than SIX MONTHS from the mailing date of this final action.

Prior Art's Arguments – Rejections

2.   Applicant's arguments filed October 16th, 2006, in particular on pages 12-14,

have been fully considered but they are not persuasive. For example,

(A) In response to applicant's argument that the references fail to show certain

features of applicant's invention, it is noted that the features upon which applicant relies

(i.e., "user-specified number of additional cycles of execution time") see response (page

12, last paragraph) are based on Applicant's interpretation of such instant limitation

upon which applicant relies ("performance of code constraints") see response (page 13,

second paragraph), Applicant specifically argues that the instant limitation relates to the

"extra execution time of the processor caused by the addition of the power down

instructions". At this point, it should be noted that this interpretation is not recited in the

rejected claim(s). As clarified in previous rejections, the plain language of the claims

merely recite "performance of code constraints". As defined in the originally filed

disclosure (See specification, page 12, lines 10-13), "The user-specified real-time

constraints can include constraints such as the number of power down instructions that

can be inserted in an execution path, the number of additional cycles of execution time

the user is willing to incur, and other such constraints".

That is to say that the aforementioned code *performance constraints* as

previously amended and herein argued by Applicant, (i.e. reduce power by constraining

execution instructions, execution time and other such constraints) are not recited in the

claims. Although the claims are interpreted in light of the specification, limitations from

the specification are not read into the claims. See *In re Van Geuns*, 988 F.2d 1181, 26

USPQ2d 1057 (Fed. Cir. 1993). Therefore, the claim limitation may be reasonably

broadly interpreted to read on examiner's position (i.e. constraining the code based on

execution time constraints) and do not necessarily require additional cycles of execution

time relating to pending claim 1 and to the extra execution time of the processor caused

by the addition of the power down instructions as expressly argued by Applicant (see

response, page 12, last paragraph). To further clarify, "performance constraints" is

applied to **Bartley** and **Li's** teachings, comprising relating power down instructions to

execution time constraints. Accordingly, the rejection of claim 1 and independent claims

14, 24 and 34 are maintained in view of Applicant's instant argument.

    (B) In response to Applicant's argument (page 13, third paragraph) reiterating the

prior art arguments from the previous response (response, dated April 19th, 2006)

regarding the rejection of the claims, for completeness. The examiner noticed that the

previous arguments were interlaced with new clarification and/or additional arguments.

As such, the examiner for the sake of completeness responds to Applicant's arguments

below for completness.

    (C) In response to applicant's argument that **Bartley** does not relate to code

performance, (page 14 of the amendment and response, third paragraph), the

examiner respectfully disagrees. It should be noted that Applicant defines user

specified real-time constraints as follows (See specification, page 12, lines 10-13),

"The user-specified real-time constraints can include constraints such as the number

of power down instructions that can be inserted in an execution path, the number of

additional cycles of <u>execution time</u> the user is willing to incur, and <u>other such</u>

<u>constraints</u>" and not solely from what is being presented ("extra execution time of the

processor caused by the addition of the power down instructions"). That is to say

that "reduced power" from the aforementioned code *performance* constraints as

being argued is a result from <u>constraining execution instructions, execution time and</u>

<u>other such constraints</u> (emphasis added).

Bartley discloses "In the case of either a compiler or assembler, an optimizing

process finds, for each functional unit, program segments during which the

functional unit is not used are located. The said segments would be of *longer*

duration than some *predetermined threshold,* wherein the processor instructions to

be inserted are based on the duration of non-activity (longer duration than some

predetermined threshold). Herein the predetermined threshold obviously relates to

execution time in order to be effective or have any meaningful result. Once these

segments are found, the compiler then inserts a power-modifying instruction at the

point in the code when the functional unit first goes out of use." (Column 7, lines 42-

43). This section of **Bartley**, clearly teaches modifying code depending on <u>time</u>

<u>constraints</u> related to execution time in order to reduce power consumption.

Although the applied passage does not expressly recite "the number of additional

cycles of <u>execution time</u>", the duration of execution time equals the additional cycles

of execution when measured in time. As such, the examiner deems that the

teaching herein is reasonably interpreted in light of the instant limitation

("*performance of code constraints*") as defined in the specification above, particularly

in light of "*other such constraints*" included in the specification definition. Therefore,

Bartley's teaching of an execution-time threshold would have been sufficient

motivation to one of ordinary skill in the art, at the time the invention was made to

consider execution time in relation to instructions to save power. Thus, the rejection

is maintained in light of the instant argument.

(D) **Li** discloses section 3.3, "Software Energy and Performance Model",

wherein the execution time of the program and the number of instructions in the

program are disclosed to directly affect the software performance model. **Li** further

teaches "Goal II: minimized power under performance constraints" (See **Li**, page 4,

Section 4.3 "System-Level Energy Optimization Algorithm". Thus, it would have

been obvious from **Li** and **Bartley's** teachings to satisfy user specified real-time

performance constraints while inserting power down instructions to reduce the power

consumption as claimed in the independent claims. Therefore, the examiner

maintains the position that **Bartley** and **Li** are reasonably pertinent to the particular

problem with which the applicant was concerned, namely energy conservation and

the rejection is maintained in light of the amendments.

Additionally, in respect to Applicant's arguments pertaining to the motivation to

combine **Bartley** and **Li** (see response, page 15, second paragraph), the examiner

disagrees. Applicant argues:

> "The purported motivation is in the context of finding code
> segments of long enough duration to make it worth shutting down a
> functional unit. If it would take longer that the amount of time required
> for execution of the segment to turn it off and then back on, it would not

make sense to turn it off in the first place. "<u>Various power modeling</u>
<u>techniques</u> can be used to <u>determine the length of time</u> during which it
is <u>more efficient</u> to turn a component off (or partially off) then on again
versus leaving it on." [see Bartley] Col. 7, lines 16-19. It does not relate
directly to satisfying user-specified real time constraints or program
performance as currently claimed.

Herein, as quoted by Applicant, the **Bartley** passage expressly discloses

"*various power modeling techniques*" to "*determine the length of time during which it*

is more efficient" to turn a component off and on. At this point again, it is stressed by

the examiner that this is taking place via power down instructions (software) in a

hardware platform or better known when combined (software/hardware) as a

system, in which, one of ordinary skill in the art is well aware that during system

engineering, particularly embedded systems, power optimization is a main concern.

As such, one of ordinary skill in the art would be motivated to look to embedded

systems (**Li**) when dealing with power optimization of any kind of software/hardware

system (**Bartley**). Thus, one of ordinary skill in the art would have been motivated to

consider performance optimization goals when dealing with system engineering

such as **Bartley**.

Applicant then contends that **Bartley** does not relate directly to program

performance (page 15, second paragraph) which examiner strongly disagrees. As

disclosed above in Applicant's quote of **Bartley**, duration of time (time constraint) to

determine software power optimization instructions placement (program code) is a

direct relation between program instruction performance and time constraints.

Again, it should be noted that **Li** is applied to the limitation of real-time performance.

(E) In regard to Applicant's arguments that the examiner's statement that the

threshold inherently must be predetermined or user specified is an alternative way of

determining the threshold (page 16, second paragraph) and as such shows the

result does not necessarily flow from the cited language, and the rejection is

improper, the examiner again strongly disagrees. It should be noted that Applicant

mischaracterizes examiners statement. Examiner intended to equate user specified

as predetermined rather than imply that one or the other may be used in an alternate

way as Applicant interpreted. To further clarify, in order to have a time duration

threshold, there necessarily must be a predetermined threshold. Now, whether a

programmer (user) chooses a known algorithm to apply, or requests a duration from

a user, there must be a specification made by a user. Therefore, a user

specification must exist in order for a predetermined threshold to exist. As such it is

inherent, as illustrated from the simple logic above, that a predetermined threshold

as disclosed by **Bartley,** was <u>user specified</u>. Therefore, the rejection is maintained

in light of the instant argument.

(F) In regard to Applicant's argument that the threshold appears to be fixed

and based on efficiency, and thus is not specified by the user as a constraint (page

16, third paragraph), the examiner refers Applicant to the above section. It is

unclear to the examiner how Applicant equates a fixed threshold or fixed time

duration in this case based on execution time to not be a constraint. A time duration

specified (predetermined time threshold) seems to clearly be a time constraint.

Accordingly, the threshold determined by the user is a user specified constraint.

(G) Independent claims 14, 24 and 34 are rejected for the reasons stated

above, as Applicant relies on the same argument as noted above. Thus, Claims 2-

10, 15-21, 25-31 and 35-42 are also rejected for at least the reason that they are

dependent on a rejected base claim.

Claim Rejections

Claims 1-44, are pending claims, and stand finally rejected in light of the

additional clarifications provided and/or addressed at item 2 above, Prior Art's

Arguments – Rejections and as provided below for completeness.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3.   Claims **1, 2, 11-15, 22-25, 32-36, 43** and **44** are rejected under 35 U.S.C. 103(a)

as being unpatentable over Bartley, US 6,219,796 (hereinafter **Bartley**), in view of Y. Li

et al. A framework for estimating and minimizing energy dissipation of embedded

hw/sw systems, (hereinafter **Li**).

In regard to claim **1**, **Bartley** discloses:

- *"A method of compiling computer code including power-down instructions to reduce power consumption during execution of the code..."* (E.g., see Figure 7 & Column 2, lines 62-67), wherein it is inherent that the code is efficient when executed by a processor.

- *"...identifying one or more potential locations in the computer code where the power-down instructions can be inserted..."* (E.g., see Figure 7 & Column 7, lines 10-21), wherein the potential locations are identified by scanning the code.

- *"...selecting locations to insert the power-down instructions from the identified potential locations in the code based on reducing power consumption ..."* (E.g., see Figure 7 & Column 7, lines 39-43), wherein the locations are determined by a predetermined threshold duration of non-use.

- *"...inserting the power-down instructions in the selected locations to reduce the power consumption during the execution of the code ..."* (E.g., see Figure 7 & Column 7, lines 43-46), wherein the power modifying or power-down instruction is then inserted to reduce the power consumption.

But **Bartley** does not expressly disclose "...*satisfying user-specified real-time constraints...*". However, **Li** discloses:

- "...*satisfying user-specified real-time performance constraints...*" (E.g.,

  see Figure 5 & Page 4, Section 4.3), wherein the user specifies one of

  many multiple objective optimization goals via performance

  constraints.

Bartley and **Li** are analogous art because they are both concerned with the

same field of endeavor, namely, an optimizing compiler with the means to reduce power

or energy consumption. Therefore, at the time the invention was made, it would have

been obvious to a person of ordinary skill in the art to combine user specified real-time

constraints with **Bartleys'** power reduction methods. The motivation is disclosed by

Bartley, as he refers to program segments having a duration longer than a

"predetermined threshold." (Column 7, lines 42-43), wherein it is obvious the threshold

may be determined by a user either via a user selected algorithm or other user input.

In regard to claim **2**, the rejections of base claim **1** are incorporated.

Furthermore, **Bartley** discloses:

- "*...wherein the code is written for a microprocessor having distinct

  functional units.*" (E.g. see Figure 7 & Column 3, lines 3-8) wherein the

  common characteristic is any processor or microprocessor that has

  more than one independent or distinct functional units.

In regard to claim **11**, the rejections of base claim **1** are

incorporated.Furthermore, **Li** discloses:

- "*... the number of power-down instructions that can be inserted in an

  execution path, including one or more identified potential locations.*"

(E.g. see Table 2 & Section 5.2), wherein the time improvement or a negative time improvement as a performance constraint is taught and may be used to limit the number of instructions inserted.

Therefore, at the time the invention was made, it would have been obvious to a person of ordinary skill in the art to combine **Li's** user specified real-time constraints with **Bartleys'** power reduction methods. The motivation is disclosed by **Bartley,** as he refers to program segments having a duration longer than a "predetermined threshold." (Column 7, lines 42-43), wherein it is obvious the threshold may be determined by a user either via a user selected algorithm or other user input. Furthermore, the segment is a direct relationship to **Li's** teaching of user specified performance constraint of time or execution cycles executed as a consequence of the energy savings. Additionally, **Bartley** provided the motivation for a number of power down instructions (E.g. see, Figure 5 & Column 2, line 11) wherein, it would have been obvious to one of ordinary skill in the art, to factor in particular power down instructions and the number of such instructions, based on the energy savings in relation to the overhead drawback.

In regard to claim **12**, the rejections of base claim **11** are incorporated. Furthermore, **Li** discloses.

- "...*the number of additional cycles of execution time the user is willing to incur due to an insertion of the power-down instruction at each of the identified potential locations.*" (E.g. see Table 2 & Section 5.2), wherein the "...minimum energy dissipation while not exceeding the budget of clock cycles to execute..." is taught.

Therefore, at the time the invention was made, it would have been obvious to a

person of ordinary skill in the art to combine user specified real-time constraints with

Bartleys' power reduction methods. The motivation is disclosed by **Bartley,** as he

refers to program segments having a duration longer than a "predetermined threshold."

(Column 7, lines 42-43), wherein it is obvious the threshold may be determined by a

user either via a user selected algorithm or other user input.

In regard to claim **13**, the rejections of base claim **11** and claim **12** are

incorporated. Furthermore **Bartley** discloses:

-   "...*inserting power-up instruction in the code to restore at least one*

    functional unit to a ready state powered-down by the inserted power-

    down instructions.." (E.g. see Figure 7 & Column 6, lines 8-19),

    wherein the power up instruction is inserted.

Therefore, at the time the invention was made, it would have been obvious to a

person of ordinary skill in the art to combine **Li's** user specified real-time constraints

with **Bartleys'** power reduction methods. The motivation is disclosed by **Bartley,** as he

refers to program segments having a duration longer than a "predetermined threshold."

(Column 7, lines 42-43), wherein it is obvious the threshold may be determined by a

user either via a user selected algorithm or other user input. Additionally, the segment

is a direct relationship to **Li's** teaching of user specified performance constraint of time

or execution cycles executed as a consequence of the energy savings.

As per claims **14, 15, 22** and **23,** this is a computer-readable medium version of

the claimed method discussed above, in claims **1, 2, 11** and **13,** wherein all claimed

limitations have also been addressed and/or cited as set forth above, wherein **Bartley** also discloses "a storage device and external memory" (16), (E.g. see, Figure 1 and associated text).

As per claims **24, 25, 32** and **33**, this is a computer system version of the claimed method discussed above, in claims **1, 2, 11** and **13**, wherein all claimed limitations have also been addressed and/or cited as set forth above, wherein **Bartley** also discloses a computer system (E.g. see, Figure 1 and associated text).

In regard to claim **34**, the rejections of claim **1** are incorporated. Additionally, **Bartley** discloses:

- *"A computer readable medium having a computer program including instructions for causing a computer to perform a method of selectively controlling power to different functional units of the computer, the instructions comprising..."* (E.g., see Figure 7 & Column 7, lines 10-21), wherein it is inherent that the instructions have to be on a computer-readable medium to be scanned by a computer process.

- *"...power-down instructions inserted in the computer-program in selected locations based on reducing power consumption..."* (E.g., see Figure 7 & Column 7, lines 10-21), wherein the potential locations are identified by scanning the code.

- *"...the power-down instructions in the selected locations reduce the power consumption during the execution of the code..."* (E.g., see

Figure 7 & Column 2, lines 6-13), wherein the locations are determined

by a predetermined threshold duration of non-use.

As per claims **35, 36, 43** and **44,** the base claim **34** is incorporated. Furthermore,

this is another computer-readable medium version of the claimed method discussed

above, in claims **1, 2, 11** and **13,** wherein all claimed limitations have also been

addressed and/or cited as set forth above, (E.g. see Figure 1 & associated text),

wherein a computer readable medium is shown (16).

4.   Claims **3-10, 16-21, 26-31** and **37-42** are rejected under 35 U.S.C. 103(a) as

being unpatentable over **Bartley** in view of **Li** and further in view of G. Ramalingam.

Data Flow Frequency Analysis, SIGPLAN Conference on Programming Language

Design and Implementation, 1996, (hereinafter **Ramalingam**). ·

In regard to claim **3**, the rejections of base claim **2** are incorporated. Furthermore,

Bartley discloses:

  - *"... based on the functional units not being used in the potential*

     locations, wherein the functional units not being used are determined

     based on functional unit usage ..." (E.g. see Figure 7 & Column 7,

     lines 10-21), wherein the functional units are not used.

But **Bartley** does not specifically disclose a *"... transfer functions at each of the*

potential locations as specified in standard monotone data-flow frameworks." However,

Ramalingam discloses:

- *"...transfer functions at each of the potential locations as specified in*

    standard monotone data-flow frameworks." (E.g. see Section 3, The

    expected Frequency of Dataflow Facts), wherein the use of transfer

    functions as specified in standard monotone data-flow frameworks is

    taught.

The combined teaching and **Ramalingam** are analogous art because they are

both concerned with the same field of endeavor, namely program optimization via

standard analysis. Therefore, at the time the invention was made, it would have been

obvious to a person of ordinary skill in the art to combine a transfer function with static

analysis method disclosed by the combined art of an optimizing compiler embodiment.

The motivation is disclosed by **Bartley,** "Locating program segments during which a

functional unit is not used may be done by either static or dynamic program analysis."

(Column 7, lines 47-49).

In regard to claim **4**, the rejections of base claim **3** are incorporated. Furthermore,

Bartley discloses:

- *"... statically analyzing processor cycles prior to executing the code."*

    (E.g. see Figure 7 & Column 7, lines 47-52), wherein the processor or

    execute cycles are estimated by the compiler for static analysis.

In regard to claim **5**, the rejections of base claim **4** are incorporated. Furthermore,

Bartley discloses:

- "...*the text in the code...*" (E.g. see Figure 7 & Column 7, lines 47-52),

    wherein the start and stop points exist in the program segments or text

    in the code.

In regard to claim **6**, the rejections of base claim **3** are incorporated.

Furthermore, **Bartley** discloses:

- "...*a first power-down instruction operable to reduce power to all of the*

    at least one functional unit, such that the functional unit is placed in a

    low state of readiness and a second power-down instruction operable

    to reduce power to only a part of the at least one functional unit, such

    that the functional unit is placed in an intermediate state of readiness."

    (E.g. see Figure 6 & Column 6, line 60 – Column 7, line 3), wherein the

    "less ready" or low state and a "more ready" or intermediated state of

    readiness are taught.

In regard to claim **7**, the rejections of base claim **1** are incorporated. But Bartley

does not expressly disclose "...*executing the code to generate power-profiling and*

execution path-profiling information..." or "...*assigning a weight factor based on the*

profile information...". However, **Li** discloses:

- "...*executing the code to generate power-profiling information*

    associated with each of the identified potential locations..." (E.g. see

    Figure 2 & Page 3, Section 3.4), wherein Figure 2 shows the program

    execution trace which generates the software performance model and

    the software energy model is also generated based on the execution

trace and then coupled with the memory energy models to account for

the total system energy generating power information or a power-

profile.

- "...*assigning a weight factor to each of the identified potential locations*

  based on the generated power-profiling..." (E.g. see Figure 5 & Section

  4.2), wherein the EES/CSI ratio or weight factor prioritizes and then

  gets assigned a probability based on the ratio. Further the EES/CSI

  numbers are based on the profile information. Additionally, the user

  specifies constraints to be met in real-time in section 4.3.

But the combined teaching of **Bartley** and **Li** do not expressly disclose

"...executing the code to generate path-profiling information...". However, **Ramalingam**

discloses:

- "...*path-profiling information...*" (E.g. see Section 1), wherein the path-

  profiling information is used to estimate probability.

- "...*and path-profiling information; and selecting the locations to insert*

  the power-down instruction from the identified locations based on the

  assigned weight factors..." (E.g. see Section 3, lemma 2), wherein the

  result is "...weighted...".

Therefore, at the time the invention was made, it would have been obvious to a

person of ordinary skill in the art to combine power and path profile information with

Bartleys' power reduction methods. Motivation was provided by **Bartley,** when he

referred to static and dynamic analysis utilizing execution cycles, loop cycles and other

"statistical predictions." (Column 7, lines 47-52), wherein it would have been obvious, at

the time the invention was made, that **Li's** constraints and profile algorithm would be

beneficial to the efficiency of a power reduction embodiment disclosed by **Bartley**.

Furthermore, motivation was provided by **Li** (Figure 2) wherein, the program execution

trace used by **Li** would only been beneficial if there was a probability that the path will

actually be used.

In regard to claim **8**, the rejections of base claim **7** are incorporated.

Furthermore, **Li** discloses:

- "...*generating execution probability of each of the identified potential*

  locations based on the generated path-profiling information." (E.g. see

  Section 3, lemma 2), wherein the result is "...weighted..." by the

  probability of execution of the path.

Therefore, at the time the invention was made, it would have been obvious to a

person of ordinary skill in the art to combine probability derived from path profile

information with **Bartleys'** power reduction methods in order to increase the efficiency

by increasing the depth of the analysis.

In regard to claim **9**, the rejections of base claim **8** are incorporated.

Furthermore, **Li** discloses:

- "...*extracting potential energy savings for each of the identified*

  potential locations using the generated power profile analysis

  information..." (E.g. see Figure 5 & Page 4, Section 4.2), wherein the

  EES is the estimated energy savings.

- *"...assigning the weight factor to each of the identified potential*

  locations based on the extracted potential energy savings and the

  generated execution probability." (E.g. see Figure 5 & Page 4, Section

  4.2), wherein the EES/CSI ratio or weight factor prioritizes and then

  gets assigned a probability based on the ratio. Further the EES/CSI

  numbers are based on the program execution trace or generated path-

  profiling information.

Therefore, at the time the invention was made, it would have been obvious to a

person of ordinary skill in the art to combine potential energy savings derived from

power profile information with **Bartleys'** power reduction methods in order to increase

the efficiency by increasing the depth of the analysis.

In regard to claim **10**, the rejections of base claim **9** are incorporated.

Furthermore, **Li** discloses:

- *"...executing the code to assign a first weight factor based on the*

  extracted potential energy savings to each of the identified potential

  locations..." (E.g. see Figure 2 & Column 3, lines 3-8), wherein the

  software performance model includes the product of execution cycles

  of a given instruction and the number of times an instruction is used or

  path profile and power information is factored to derive a weight factor.

- *"... executing the code to assign a second weight factor based on*

  execution probability at each of the identified potential locations..."

  (E.g. see Figure 2 & Column 3, lines 3-8), wherein the software

performance model includes the product of execution cycles of a given

instruction and the number of times an instruction is used or path

profile.

-    "...*computing a product of the first and second weight factors for each*

of the identified potential locations; calculating the weight factor for

each of the identified potential locations based on the computed

product of the first and second weight factors; and assigning the

calculated weight factor to each of the identified potential locations."

(E.g. see Figure 2 & Column 3, lines 3-8), wherein the software

performance model includes the product of execution cycles of a given

instruction and the number of times an instruction is used or path

profile and the weight factor is assigned based on a product of

weighted factors of both the energy savings or power profile and

execution probability. The EES/CSI ratio as disclosed above is based

on the products of path and profile information.

As per claims **16-21,** this is a computer-readable medium version of the claimed

method discussed above, in claims **3, 4** and **7-10,** wherein all claimed limitations have

also been addressed and/or cited as set forth above, (E.g. see Figure 1 & associated

text), wherein a computer readable medium is shown (16).

As per claims **26-31,** this is a computer system version of the claimed method

discussed above, in claims **3, 4** and **7-10,** wherein all claimed limitations have also been

addressed and/or cited as set forth above, (E.g. see Figure 1 & Column 3, lines 3-8),

wherein a computer system is shown.

As per claims **37-42,** the base claim **34** and **35** are incorporated. Furthermore,

this is another computer system version of the claimed method discussed above, in

claims **3, 4** and **7-10,** wherein all claimed limitations have also been addressed and/or

cited as set forth above, (E.g. see Figure 2 & Column 3, lines 3-8), wherein a computer

system is shown.

Conclusion

Any inquiry concerning this communication or earlier communications from the

examiner should be directed to John J. Romano whose telephone number is (571) 272-

3872. The examiner can normally be reached on 8-5:30, M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number

for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the

Patent Application Information Retrieval (PAIR) system. Status information for

published applications may be obtained from either Private PAIR or Public PAIR.

Status information for unpublished applications is available through Private PAIR only.

For more information about the PAIR system, see http://pair-direct.uspto.gov. Should

you have questions on access to the Private PAIR system, contact the Electronic

Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a

USPTO Customer Service Representative or access to the automated information

system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

JJR

TUAN DAM
SUPERVISORY PATENT EXAMINER